AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the complete listing of claims listed below. This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for execution by a microprocessor in response to receiving a single instruction, the method comprising:

receiving a string of bits;

generating a plurality of indices using a plurality of segments of bits in the string of bits;

looking up simultaneously a plurality of entries from a plurality of look-up tables using the plurality of indices, wherein each of said plurality of look-up tables is separate and distinct from others of said plurality of look-up tables; and

combining the plurality of entries into a first result;

wherein the above operations are performed in response to the microprocessor receiving the single instruction.

2. (Original) A method as in claim 1 further comprising:

receiving a plurality of data elements specifying the plurality of segments in the string of bits.

3. (Original) A method as in claim 2 wherein the plurality of data elements are received from an entry in a register file; and wherein the microprocessor is a media processor integrated with a memory controller on a single integrated circuit.

4. (Original) A method as in claim 3 wherein the single instruction specifies an index of the entry in the register file.

5. (Original) A method as in claim 2 further comprising:
receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements.

6. (Original) A method as in claim 5 further comprising:
generating a new bit pointer using the first result.

7. (Original) A method as in claim 1 further comprising:
receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits.

8. (Original) A method as in claim 1 further comprising:
partitioning look-up memory into the plurality of look-up tables before said looking-up;
wherein the microprocessor is a media processor formed on a monolithic integrated circuit.

9. (Original) A method as in claim 8 wherein the look-up memory comprises a plurality of look-up units, and wherein said partitioning look-up memory comprises:
configuring the plurality of look-up units into the plurality of look-up tables.

10. (Previously Presented) A method as in claim 9 wherein each of the plurality of look-up units comprises 256 8-bit entries.

11. (Original) A method as in claim 1 wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables.

12. (Original) A method as in claim 11 wherein the total number of entries is one of:

a) 256;

b) 512; and

c) 1024.

13. (Original) A method as in claim 1 wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables.

14. (Original) A method as in claim 13 wherein the total number of bits is one of:

a) 8;

b) 16; and

c) 24.

15. (Previously Presented) A method as in claim 8 wherein the plurality of look-up tables are configured according to an indicator in an entry in a register file.

16. (Original) A method as in claim 15 wherein the single instruction specifies an index of the entry in the register file.

17. (Original) A method as in claim 1 wherein said combining the plurality of entries comprises:

selecting a valid data from the plurality of entries.

18. (Currently Amended) A method as in claim 17 further comprising:

generating an indicator indicating whether none of the plurality of entries [[is]] are valid.

19. (Original) A method as in claim 17 wherein the valid data is selected according to priorities of the look-up tables from which the plurality of entries are looked up.

20. (Original) A method as in claim 17 wherein said combining the plurality of entries further comprises:

formatting the valid data according to a type of the valid data.

21. (Currently Amended) A method as in claim 20 wherein the type of the valid data is one of:

a) zero fill;

b) sign magnitude; and

c) [[two]] two's complement.

22. (Original) A method as in claim 21 further comprising:

retrieving a sign bit from the string of bits for the valid data,

wherein the first result is obtained by formatting the valid

data using the sign bit when the type of the valid data is sign magnitude.

23. (Original) A method as in claim 1 wherein an entry in the plurality of entries contains:

a) information indicating whether the entry is valid;

b) information indicating a type of the entry; and

c) information indicating a number of bits of a code word to be decoded.

24. (Original) A method as in claim 1 wherein the string is received from an entry in a register file.

25. (Original) A method as in claim 24 wherein the single instruction specifies an index of the entry in the register file.

26. (Currently Amended) A method as in claim 1 further comprising:

receiving a first number indicating a position of a last bit of input in the string of [[bit]] bits.

27. (Original) A method as in claim 26 further comprising:

generating an indicator indicating whether any bit after the last bit of input is used in obtaining the first result.

28. (Original) A method as in claim 12 further comprising:

generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code.

29. (Original) A method as in claim 28 wherein the predetermined code represents an end of block condition.

30. (Original) A method as in claim 1 further comprising:
receiving at least one format;
formatting the string of bits into at least one escape data according to the at least one format; and
combining the at least one escape data and the first result into a second result.

31. (Currently Amended) A method as in claim 30 wherein one of the at least one format is for data of a type which is one of:
a) zero fill;
b) sign magnitude; and
c) [[two]] two's complement.

32. (Original) A method as in claim 30 wherein the at least one format is received from an entry of a register file.

33. (Original) A method as in claim 32 wherein the single instruction specifies an index of the entry in the register file.

34. (Currently Amended) A machine readable media containing an executable computer program instruction which when executed by a digital processing system causes said system to perform a method comprising:
receiving a string of bits;

generating a plurality of indices using a plurality of segments of bits in the string of bits;

looking up simultaneously a plurality of entries from a plurality of look-up tables using the plurality of indices, wherein each of said plurality of look-up tables is separate and distinct from others of said plurality of look-up tables; and

combining the plurality of entries into a first result;

wherein the above operations are performed in response to the microprocessor receiving the single instruction.

35. (Original) A media as in claim 34 wherein the method further comprises:

receiving a plurality of data elements specifying the plurality of segments in the string of bits.

36. (Original) A media as in claim 35 wherein the plurality of data elements are received from an entry in a register file.

37. (Original) A media as in claim 36 wherein the single instruction specifies an index of the entry in the register file.

38. (Original) A media as in claim 35 wherein the method further comprises:

receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements.

39. (Original) A media as in claim 38 wherein the method further comprises:

generating a new bit pointer using the first result.

40. (Original) A media as in claim 34 wherein the method further comprises:
receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits.

41. (Original) A media as in claim 34 wherein the method further comprises:
partitioning look-up memory into the plurality of look-up tables before said looking-up.

42. (Original) A media as in claim 41 wherein the look-up memory comprises a plurality of look-up units, and wherein said partitioning look-up memory comprises:
configuring the plurality of look-up units into the plurality of look-up tables.

43. (Previously Presented) A media as in claim 42 wherein each of the plurality of look-up units comprises 256 8-bit entries.

44. (Original) A media as in claim 34 wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables.

45. (Original) A media as in claim 44 wherein the total number of entries is one of:
a) 256;
b) 512; and
c) 1024.

46. (Original) A media as in claim 34 wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables.

47. (Original) A media as in claim 46 wherein the total number of bits is one of:

a) 8;

b) 16; and

c) 24.

48. (Previously Presented) A media as in claim 41 wherein the plurality of look-up tables are configured according to an indicator in an entry in a register file.

49. (Original) A media as in claim 48 wherein the single instruction specifies an index of the entry in the register file.

50. (Original) A media as in claim 34 wherein said combining the plurality of entries comprises:

selecting a valid data from the plurality of entries.

51. (Currently Amended) A media as in claim 50 wherein the method further comprises:

generating an indicator indicating whether none of the plurality of entries [[is]] <u>are</u> valid.

52. (Original) A media as in claim 50 wherein the valid data is selected according to priorities of the look-up tables from which the plurality of entries are looked up.

53. (Original) A media as in claim 50 wherein said combining the plurality of entries further comprises:

formatting the valid data according to a type of the valid data.

54. (Currently Amended) A media as in claim 53 wherein the type of the valid data is one of:

a) zero fill;

b) sign magnitude; and

c) [[two]] two's complement.

55. (Original) A media as in claim 54 wherein the method further comprises:

retrieving a sign bit from the string of bits for the valid data, wherein the first result is obtained by formatting the valid data using the sign bit when the type of the valid data is sign magnitude.

56. (Original) A media as in claim 34 wherein an entry in the plurality of entries contains:

a) information indicating whether the entry is valid;

b) information indicating a type of the entry; and

c) information indicating a number of bits of a code word to be decoded.

57. (Original) A media as in claim 34 wherein the string is received from an entry in a register file.

58. (Original) A media as in claim 57 wherein the single instruction specifies an index of the entry in the register file.

59. (Currently Amended) A media as in claim 34 wherein the method further comprises:

receiving a first number indicating a position of a last bit of input in the string of [[bit]] bits.

60. (Original) A media as in claim 59 wherein the method further comprises:

generating an indicator indicating whether any bit after the last bit of input is used in obtaining the first result.

61. (Original) A media as in claim 45 wherein the method further comprises:

generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code.

62. (Original) A media as in claim 61 wherein the predetermined code represents an end of block condition.

63. (Original) A media as in claim 34 wherein the method further comprises:

receiving at least one format;

formatting the string of bits into at least one escape data according to the at least one format; and

combining the at least one escape data and the first result into a second result.

64. (Currently Amended) A media as in claim 63 wherein one of the at least one format is for data of a type which is one of:

a) zero fill;

b) sign magnitude; and

c) [[two]] two's complement.

65. (Original) A media as in claim 63 wherein the at least one format is received from an entry of a register file.

66. (Original) A media as in claim 65 wherein the single instruction specifies an index of the entry in the register file.

67. (Previously Presented) A method as in claim 1 wherein the plurality of segments of bits in the string of bits are of variable lengths.

68. (Previously Presented) A method as in claim 1 wherein each of the plurality of indices corresponds to a different one of the plurality of look-up tables.

69. (Previously Presented) A method as in claim 1 wherein each of the plurality of look-up tables is larger than a vector register.